Risk-Aware Multi-Objective Optimization for AI Infrastructure Supply Allocation under Demand and Supplier Uncertainty

Suchuan Xing[1], Yihan Wang[2,*]

[1]Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA

[2]School of Engineering and Applied Science, The University of Pennsylvania, Philadelphia, PA 19104, USA

[*]Corresponding email: wangyh1@alumni.upenn.edu

Abstract

Operators of AI infrastructure must commit accelerator procurement months before demand is known, from a supply base whose deliveries are themselves unreliable. Existing supply-allocation models optimize expected cost and treat service risk and environmental impact as constraints or afterthoughts, which hides the trade-off that actually drives capital decisions. This paper formulates AI-accelerator supply allocation as a two-stage risk-averse three-objective stochastic program that minimizes expected total cost, the conditional value-at-risk (CVaR) of the service-level shortfall, and expected carbon emissions, subject to minimum order quantities, all-units volume discounts, capital budgets and diversification limits, under jointly uncertain demand, fulfilment yield, correlated regional disruptions and lead-time slippage. Demand scenarios are generated from a moving-block bootstrap of a real two-month production GPU-cluster trace comprising 737,137 tasks. We solve the problem with RA-TGEA, a matheuristic that combines (i) an exact closed-form vectorized recourse evaluator, verified against linear programming to a maximum relative error of 1.9e-14, (ii) a direction-aware local search driven by an exact closed-form sub-gradient of the CVaR tail, and (iii) progressive scenario refinement with tail-preserving stratified reduction. On an exactly solvable instance, RA-TGEA attains 98.1% of the hypervolume of an epsilon-constraint MILP reference front while being roughly three orders of magnitude faster at producing a dense front. Against five state-of-the-art multi-objective evolutionary algorithms under an equal wall-clock budget, it improves hypervolume by 9.1% and 20.4% on the two larger instances (p = 0.014). Out of sample on 2,000 unseen scenarios, the risk-aware model reduces attainable tail shortfall by up to 36.1% relative to a deterministic expected-value plan at the same cost. A sensitivity study quantifies the price of resilience: halving the tail shortfall target from 0.10 to 0.05 costs 56 percentage points of additional expenditure.

Keywords

AI infrastructure, supply allocation, conditional value-at-risk, multi-objective optimization, stochastic programming, supply chain resilience, sustainable computing.

1. Introduction

The build-out of machine-learning infrastructure has turned accelerator procurement into a first-order strategic problem. Training compute has grown by orders of magnitude over the last decade [1], and production clusters now aggregate thousands of heterogeneous GPUs serving a mixture of large distributed training jobs, ordinary training jobs and latency-sensitive inference tasks [2]-[4]. Because a purchase commitment precedes delivery by months, and because

delivered capacity is durable, the allocation of orders across a supply base is a capital decision taken under two simultaneous sources of uncertainty: how much compute will be demanded, and how much of what was ordered will actually arrive on time.

Both sources are severe in this domain. On the demand side, growth rates rather than short-term fluctuations dominate the planning risk. On the supply side, accelerator supply chains concentrate on a handful of vendors and foundries, so vendor-specific yield shortfalls, regionally correlated disruptions and lead-time slippage are the norm rather than the exception; the operations-research literature on supply disruptions has documented the value of mitigation, diversification and contingency strategies in exactly this regime [5]-[9]. A third consideration has become unavoidable: the carbon footprint of a procurement decision is largely locked in at the moment of purchase, through both the embodied emissions of the devices and the grid intensity of the region in which they are deployed [10]-[14].

These three concerns — cost, service risk and carbon — genuinely conflict, and the conflict is not resolved by optimizing expected cost subject to service and emission constraints, because the decision maker generally does not know the right constraint levels a priori. What a planner needs is the attainable trade-off surface, together with an explicit treatment of the tail of the service distribution: a plan whose expected shortfall is acceptable may still fail catastrophically in the 5% of futures in which demand grows fast and a major supplier is disrupted simultaneously.

This paper makes the following contributions.

1) Formulation. We formulate AI-accelerator supply allocation as a two-stage risk-averse three-objective stochastic mixed-integer program. The objectives are expected total cost, the CVaR of a weighted service-level shortfall index, and expected carbon emissions. The first stage decides supplier qualification, per-period order quantities and an operating-policy parameter; the second stage allocates realized capacity to service classes and rents on-demand capacity. The model retains features that matter in practice and destroy convexity: minimum order quantities, all-units volume discounts, per-period capital budgets, supplier-share caps and a minimum number of qualified suppliers.

2) Trace-driven uncertainty. Demand scenarios are produced by a moving-block bootstrap of period-over-period demand levels extracted from a real production GPU cluster trace (Alibaba PAI, two months, 737,137 GPU tasks) [2], combined with an explicit scenario-wise growth-rate factor. Supplier uncertainty combines Beta fulfilment ratios, regionally correlated disruptions with random duration, and stochastic lead-time slippage.

3) Solution method. We propose RA-TGEA (Risk-Aware Tail-Gradient Evolutionary Algorithm). Its three components are an exact closed-form recourse evaluator that removes the inner linear program entirely; a direction-aware local search driven by an exact closed-form sub-gradient of the CVaR tail with respect to order quantities; and progressive scenario refinement with tail-preserving stratified scenario reduction, whose importance weights keep the reduced-set estimators of both the expectation and the CVaR unbiased.

4) Validation and evidence. The closed-form recourse is verified against a linear-programming solution of the same subproblem, and the tail sub-gradient against finite differences. An epsilon-constraint MILP provides an exact reference front on a small instance. On larger instances RA-TGEA outperforms NSGA-II, NSGA-III, MOEA/D, SMS-EMOA and C-TAEA under an equal wall-clock budget, and out-of-sample experiments quantify the value of risk-aware over deterministic planning.

2. Related Work

A. Risk-averse supply allocation

Supplier selection and order allocation under disruption risk is a mature field. Snyder et al. [5] survey the operations-research models; Tomlin [6] characterizes when mitigation dominates contingency. Sawik [7]-[9] develops portfolio formulations in which value-at-risk and CVaR control the cost tail, solved as mixed-integer programs. Risk-averse two-stage stochastic programming with coherent risk measures [15]-[17] provides the theoretical basis; CVaR is attractive because it is coherent and admits the Rockafellar-Uryasev linearization [15], [16]. Robust and distributionally robust alternatives trade distributional information for worst-case guarantees [18], [19]. Our formulation departs from this literature in three ways: the risk measure is applied to a service index rather than to cost, so that risk and cost remain separate objectives; carbon is a third objective rather than a constraint; and the uncertainty description is estimated from a real workload trace rather than assumed.

B. AI infrastructure and its footprint

Characterization studies of production GPU clusters [2]-[4] document the workload heterogeneity we exploit to define service classes, and they release the traces that make data-driven scenario generation possible. On the environmental side, Gupta et al. [11], [12] show that embodied emissions are comparable to operational emissions for modern computing hardware and provide architectural carbon models; Patterson et al. [13] and Masanet et al. [14] quantify the strong dependence of operational emissions on deployment region. Semiconductor capacity planning under uncertainty [20], [21] is the closest classical analogue to our setting, but it addresses fab capacity rather than the buyer-side allocation problem.

C. Multi-objective evolutionary optimization

We build on reference-vector-guided selection in the style of NSGA-III [22] with constrained non-dominated sorting [23], simulated binary crossover and polynomial mutation [24], and Das-Dennis reference directions [25]. NSGA-II [23], MOEA/D [26], SMS-EMOA [27] and C-TAEA [28] serve as baselines, implemented in pymoo [29]. Quality is measured by hypervolume [30] and IGD+ [31]. Evolutionary algorithms have been applied to stochastic supply-chain design before; the novelty here lies in exploiting the analytic structure of the risk-averse recourse to make each fitness evaluation cheap and to make variation risk-aware.

3. Problem Formulation

A. Setting and notation

A planner allocates orders for accelerator capacity over T procurement periods across N candidate suppliers, to serve K service classes. All monetary quantities are expressed in normalized monetary units (m.u.), one m.u. being the acquisition price of one baseline accelerator. Supplier s is characterized by a list price c1(s) and a discounted price c2(s) that applies to the whole volume once cumulative commitments reach a threshold B(s); a qualification cost Fx(s); a minimum order quantity m(s); a per-period allocation cap U(s); a nominal lead time L(s) periods; a mean fulfilment ratio r(s); a marginal disruption probability pi(s); a region g(s); an effective capacity factor phi(s); an energy cost e(s) and an operational carbon factor ce(s) per installed unit-period; and an embodied carbon factor cb(s) per delivered unit. Class k has an SLA penalty pn(k) per unit-period of unmet demand and an SLA weight w(k). An on-demand market supplies up to a fraction of period demand at price c_spot per unit-period.

B. First-stage decisions and constraints

The first stage chooses order quantities q(s,t) >= 0, the implied qualification indicators y(s) = 1{sum_t q(s,t) > 0}, and a scalar operating-policy parameter beta in [0,1] described below. The feasible set is

$$q(s,t) = 0 \;\; or \;\; m(s) <= q(s,t) <= U(s) \tag{1}$$

$$sum\_t\ q(s,t) <= rho\_max * sum\_\{s',t\}\ q(s',t) \quad (2)$$

$$sum\_s\ price(s)\ q(s,t) <= Bud(t) \quad for\ all\ t \quad (3)$$

$$sum\_s\ y(s) >= n\_min \quad (4)$$

where price(s) equals c2(s) if the cumulative commitment reaches B(s) and c1(s) otherwise. Constraint (1) is the minimum-order-quantity condition, (2) caps the share of any single supplier, (3) is a per-period capital budget and (4) enforces a minimum supply-base breadth. The all-units discount in price(s) makes the cost a discontinuous, non-convex function of q, and (1) is disjunctive; both are the reason the problem is not a linear program.

C. Uncertainty

A scenario omega specifies demand d(k,t,omega), a fulfilment ratio theta(s,t,omega) in [0,1] applied to the order placed for period t, and an arrival period a(s,t,omega). Demand is generated as

$$d(k,t,omega) = d0(k)\ (1+g(omega))^t\ eps(k,t,omega) \quad (5)$$

where d0(k) is the class base level and eps is a moving-block bootstrap draw from the empirical normalized demand levels of the real trace, so that the empirical coefficient of variation, short-run autocorrelation and cross-class correlation are preserved. The scenario growth rate g(omega) is drawn from a normal distribution with mean g and standard deviation sd_g and represents medium-term demand-growth uncertainty, which dominates the planning risk over a multi-period horizon. Fulfilment ratios are Beta variates with mean r(s). Disruptions are generated from independent supplier events with probability pi(s) and a common regional shock with probability pi_reg affecting all suppliers in a region simultaneously; a disruption covering an arrival period defers the shipment to the first period after the disruption ends and destroys half of it. Lead times slip by 0, 1 or 2 periods with fixed probabilities. Shipments arriving after the horizon are paid for but never installed.

D. Second stage

Installed capacity accumulates, because delivered accelerators keep serving in every subsequent period:

$$C(t,omega) = C0 + sum\_\{s\}\ sum\_\{t': a(s,t',omega) <= t\}\ phi(s)\ theta(s,t',omega)\ q(s,t') \quad (6)$$

Usable capacity is zeta*C(t,omega) with zeta < 1 accounting for fragmentation and maintenance. Given usable capacity, the operational response in period t minimizes realized operating cost, i.e. the sum of on-demand rental cost and SLA penalties, subject to the demand balance x + rent + unmet = d, the capacity limit sum_k x <= zeta C, and the on-demand cap. The policy parameter beta scales the willingness to rent on-demand capacity for classes whose SLA penalty does not by itself justify the rental price, which lets the planner buy tail insurance operationally rather than only through hardware.

E. Objectives

Let V(omega) be the weighted service-shortfall index, i.e. the SLA-weighted unmet demand divided by the SLA-weighted total demand of scenario omega, so V in [0,1]. The three objectives, all minimized, are

$$f1 = sum\_s\ Fx(s)\ y(s) + E[\ proc + energy + rental + penalty\ ] \quad (7)$$

$$f2 = CVaR\_alpha[\ V\ ] = min\_xi\ \{\ xi + E[(V-xi)+]\ /\ (1-alpha)\ \} \quad (8)$$

$$f3 = E[\ embodied + operational + rental\ carbon\ ] \quad (9)$$

Procurement is paid on delivered units plus a non-refundable deposit on committed-but-undelivered units. The two cost-like objectives are not redundant: f1 aggregates monetized expectations, whereas f2 is a tail functional of a normalized service metric and is invariant to how shortfall is priced. Replacing (8) by E[V] or by max V yields the risk-neutral and robust variants used as modelling baselines in Section VI-E.

F. Deterministic equivalent

With a finite scenario set the model admits a deterministic-equivalent MILP: (6) is linear in q, the recourse is a linear program, the all-units discount is modelled by splitting q into two tier variables with a binary indicator, (1) requires a per-order binary, and (8) is linearized in the Rockafellar-Uryasev form [15] with one auxiliary variable per scenario. This MILP is used only to obtain exact reference fronts, since its size grows linearly in the number of scenarios: 1,921 variables and 30 binaries at N=5, T=4, S=50, and 29,409 variables at N=8, T=8, S=400.

4. The Ra-Tgea Solver

A. Encoding and deterministic repair

An individual is a vector z in [0,1]^(NT+1): the first NT entries scale to order quantities through q(s,t) = z(s,t) U(s), and the last entry is beta. A deterministic repair maps any z to a feasible plan. It applies minimum-order-quantity snapping (orders below m(s) are set to zero), then iteratively scales down suppliers that violate the share cap (2), then scales down periods that violate the budget (3), re-snapping after each step. Every step is non-increasing in total spend, so (1)-(3) hold after repair; only (4) is left as a constraint-violation measure handled by constrained non-dominated sorting. The same repair is used by all compared algorithms, so comparisons isolate the search mechanism.

B. Exact closed-form recourse

The second-stage problem of Section III-D is, for each scenario and period, a transportation problem with two sources of class-independent cost (free installed capacity and on-demand rental at c_spot) and class-dependent dump costs pn(k). Its optimal solution is obtained by a greedy rule: serve classes in decreasing order of SLA penalty from installed capacity, then rent for the classes whose penalty exceeds c_spot in the same order until the cap binds, then rent a fraction beta of the residual cap for the remaining classes. Because installed capacity accumulates rather than being consumed, periods decouple given (6), so the whole recourse over all scenarios and periods reduces to cumulative sums and clipping and can be evaluated with dense array operations.

Proposition 1. The greedy rule attains the optimum of the second-stage linear program for beta = 0, and for beta > 0 it is a feasible policy whose objective value is an upper bound on that optimum. The first part follows from the rearrangement argument for a two-source transportation problem with source costs uniform across sinks; the second is immediate since the rule remains feasible. We verified Proposition 1 numerically on 300 randomly drawn (plan, scenario, period) triples: the maximum relative deviation between the closed-form operating cost and a linear-programming solution of the same subproblem was 1.9e-14. One full evaluation of all three objectives costs 0.24 ms at S=100 and 0.73 ms at S=400 on a single core,

which is three to four orders of magnitude cheaper than solving the corresponding scenario linear programs.

C. Direction-aware tail-gradient local search

CVaR depends only on the alpha-tail of the scenario distribution, and the greedy recourse makes the derivative of V with respect to capacity available in closed form. Let T_alpha be the set of tail scenarios, and in scenario omega and period t let k(omega,t) be the first class in penalty order with positive unmet demand. One extra unit of usable capacity in period t reduces V(omega) at rate w(k(omega,t)) zeta divided by the SLA-weighted total demand; a unit ordered from supplier s for period t contributes phi(s) theta(s,t,omega) units from its arrival period onward. Hence

$$dV(\omega)/dq(s,t) = -\,\phi(s)\,\theta(s,t,\omega)\sum_{\tau \geq a(s,t,\omega)} mg(\omega,\tau) \quad (10)$$

and averaging (10) over T_alpha gives an exact sub-gradient of the tail average at cost O(|T_alpha| N T).

Proposition 2. Expression (10) is a sub-gradient of the alpha-tail average of V with respect to q wherever the greedy allocation is locally constant, i.e. almost everywhere. Comparing (10) with central finite differences over all NT coordinates gave a correlation of 0.9998 and a mean relative deviation of 3.2%, the residual being attributable to kinks of the piecewise-linear map.

The operator divides the tail gradient by the effective unit price to obtain a tail benefit per monetary unit, then performs one of two moves with equal probability: a risk-reducing transfer, which withdraws a random fraction of volume from the cells with the lowest ratio and redistributes it to the cells with the highest ratio, or a cost-reducing withdrawal, which only removes volume from the lowest-ratio cells. The first move improves f2 at nearly constant f1; the second improves f1 and f3 at the expense of f2. Applying both keeps the operator useful across the whole reference-vector fan instead of biasing the population towards the risk-averse end. It is applied to 30% of the offspring.

D. Progressive scenario refinement

Sample-average approximation [34] makes evaluation cost linear in the number of scenarios, but early generations do not need an accurate estimate. We therefore grow the evaluation sample geometrically from S_min to the full set as the run progresses, rebuilding the reduced set every six generations. Unlike classical scenario reduction, which minimizes a probability metric between the full and reduced laws [35], our reduction is tail-preserving: scenarios are stratified by a severity index combining standardized supply loss and standardized aggregate demand, the tail stratum receives probability mass 2(1-alpha) and 55% of the reduced sample, and each retained scenario carries an importance weight equal to its stratum mass divided by the number retained from that stratum. With those weights both the expectation estimators in (7), (9) and the weighted CVaR estimator in (8) remain unbiased, while the tail that determines f2 is sampled densely. The final population and the last archives are re-evaluated on the full scenario set before being returned, so no reported objective value depends on the reduction.

E. Algorithm and complexity

RA-TGEA is an elitist reference-vector algorithm: it generates offspring by simulated binary crossover and polynomial mutation [24], applies the local search of Section IV-C to a subset of them, evaluates parents and offspring on the current scenario sample, and selects survivors by constrained non-dominated sorting followed by reference-direction niching in normalized objective space. With population size P, generations G and scenario sample S the cost is O(PG(NTS + TKS)) array operations plus O(P^2) dominance comparisons per generation, all of which are vectorized.

5. Experimental Setup

A. Real trace and demand calibration

We use the task table of the public Alibaba PAI GPU-cluster trace (cluster-trace-gpu-v2020), released with [2], covering a production cluster of more than 6,000 GPUs over two months. The archive checksum matches the value published by the maintainers. Of 1,261,050 task records, 737,137 request GPU resources and are usable; GPU demand is computed as the number of instances times the requested GPU fraction, and hourly concurrency is reconstructed from task start and end times over a 68.5-day span. Three service classes are defined from the data: C1, gang-scheduled distributed training with at least four instances and duration of at least one hour; C3, fractional-GPU or sub-15-minute latency-sensitive tasks; and C2, the remainder. A planning period aggregates 72 hours and its provisioning target is the 95th percentile of hourly concurrency inside the period, giving 20 consecutive observations per class. Table I reports the resulting statistics. They are used in three ways: the class base levels d0 set the instance scale, the normalized level deviations feed the bootstrap in (5), and the reported dispersion is what the volatility multiplier in the sensitivity study scales. A first ramp-up period is excluded from calibration.

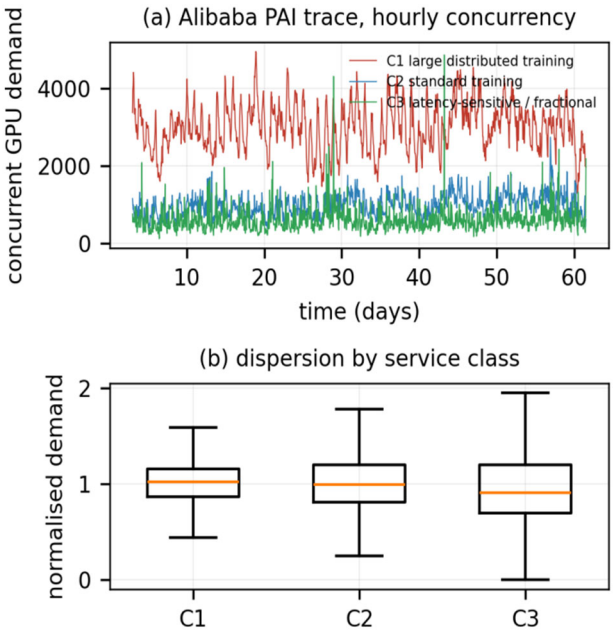

Figure 1. Demand signal extracted from the real Alibaba PAI GPU trace. (a) Hourly concurrent GPU demand per service class. (b) Dispersion of hourly demand normalized by the class mean.

Table I. Demand statistics estimated from the real GPU trace

Statistic	C1	C2	C3
Share of GPU-hours	0.783	0.186	0.031
Mean period demand (GPUs)	3905	1379	1105
CV of period demand	0.082	0.164	0.223
Autocorrelation of log growth	-0.599	-0.232	-0.473
Hourly P95 / mean	1.361	1.513	1.840

Cross-class correlations of log growth rates: (C1,C2) 0.328, (C1,C3) -0.256, (C2,C3) 0.308. 737,137 GPU tasks over 68.5 days, 20 planning periods.

B. Instances

Three instances of increasing size are used: I1 (N=6, T=6, S=300), I2 (N=8, T=8, S=400) and I3 (N=12, T=10, S=500, with tighter supplier capacities), plus a small instance I0 (N=5, T=4, S=50) on which the MILP is solved exactly. Supplier attributes are drawn once per instance from fixed seeds over ranges chosen so that price, reliability and carbon intensity are negatively correlated by construction: cheap suppliers are less reliable and sit in higher-carbon regions, which is what makes the three objectives conflict. Table II lists the ranges and their grounding. Procurement periods are months; a period-long shortfall of one accelerator is penalized at 0.20, 0.11 and 0.28 m.u. for C1, C2 and C3 respectively, and on-demand capacity costs 0.14 m.u. per unit-period, which corresponds to roughly USD 6 per GPU-hour if one m.u. is taken as USD 30,000. Unless stated otherwise alpha = 0.95, g = 0.06, sd_g = 0.03, rho_max = 0.45 and n_min = 3.

Table II. Instance parameter ranges

Parameter	Range	Basis
List price (m.u./unit)	0.82-1.24	normalized
Volume discount	4-10%	assumed
Mean fulfilment ratio	0.70-0.99	[5], [7]
Disruption probability	0.02-0.18	[5], [6]
Regional shock probability	0.05	[5]
Lead time (periods)	0-3	assumed
Power per unit (kW)	0.30-0.65	[11]-[13]
PUE	1.10-1.55	[13], [14]
Grid intensity (kg/kWh)	0.05-0.65	[13], [14]
Embodied carbon (kg/unit)	90-212	[11], [12]
Electricity price (USD/kWh)	0.06-0.16	assumed

Supplier attributes are generated so that price is negatively correlated with reliability and with carbon cleanliness.

C. Protocol

RA-TGEA is compared with NSGA-III, NSGA-II, MOEA/D, SMS-EMOA and C-TAEA from pymoo [29], all operating on the identical encoding, repair and evaluator, with population size 91 and Das-Dennis reference directions of 12 partitions where applicable. MOEA/D cannot handle constraints in this implementation, so its objectives are penalized by the constraint violation. Each algorithm receives the same wall-clock budget of 6 s per run on a single core, and each configuration is run with 4 seeds; an additional experiment equalizes the number of evaluations at 12,000 instead. Quality is measured by hypervolume and IGD+ after normalizing objectives by the ideal and nadir of the pooled non-dominated set of all algorithms and seeds on that instance, with reference point 1.1 in each normalized dimension; the IGD+ reference set is the pooled non-dominated set, or the exact MILP front for I0. Significance is assessed by one-sided Mann-Whitney U tests. All experiments run on one CPU core with 3 GB of memory, using SciPy [32] with the HiGHS MILP solver [33].

6. Results

A. Validation against an exact reference front

On I0 we build a reference front by an augmented epsilon-constraint scan: three lexicographically corrected anchor solves followed by a 6x5 grid of epsilon values on f2 and f3, minimizing f1. The 33 MILP solves take 17.0 s in total and yield 24 non-dominated points. The anchors are f1 = 568.0 m.u. at CVaR 0.154, CVaR 0.0103 at cost 4267.9 m.u., and zero procurement-related carbon at CVaR 0.204, so the three objectives span wide and genuinely conflicting ranges.

Table III compares the heuristics with this exact front. RA-TGEA reaches 98.1% of the exact hypervolume in 6.1 s and NSGA-III 99.1% in 10.0 s; the best CVaR found by RA-TGEA is within 5.1e-05 of the exact minimum in normalized terms. On this small instance the exact method is therefore competitive, and we report it as such. The picture changes with scale. Table III also reports the time to compute a single Pareto point with a binding tail-risk epsilon-constraint: 1.9 s at I0 size, but 10.7 s, 24.9 s and 63.7 s as the scenario count grows to 100, 200 and 400 at the I2 configuration, and the model reaches 36,785 variables at the I3 configuration. Producing a 91-point front at the I2 configuration by epsilon-constraint scanning would therefore require roughly 1.6 hours, against 6 s for RA-TGEA, i.e. about three orders of magnitude. This is the practical argument for the matheuristic: not that the MILP is unsolvable, but that a dense trade-off surface requires hundreds of solves.

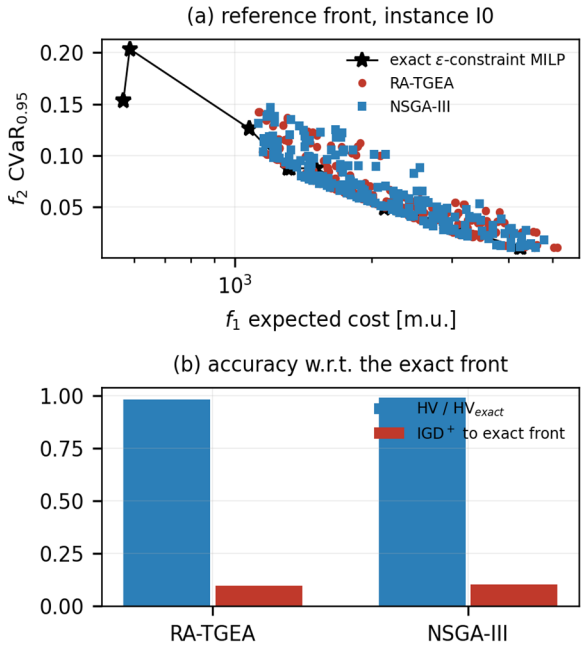

Figure 2. (a) Exact epsilon-constraint MILP reference front on instance I0 against the heuristic fronts, projected on cost and tail risk. (b) Hypervolume ratio to the exact front and IGD+ to the exact front.

Table III. Validation against the exact MILP (instance I0) and cost of exact solving at scale

Method / size	HV	HV ratio	IGD+	Time
Exact MILP front (24 pts)	0.869	1.000	0.000	17.0 s
RA-TGEA (91 pts)	0.853	0.981	0.097	6.1 s
NSGA-III (43 pts)	0.861	0.991	0.101	10.0 s
One MILP point, S=50	-	-	-	1.9 s
One MILP point, S=100	-	-	-	10.7 s
One MILP point, S=200	-	-	-	24.9 s
One MILP point, S=400	-	-	-	63.7 s

Heuristic values are means over 5 seeds. Single-point MILP times are for one epsilon-constrained solve with a binding tail-risk target at the I2 supplier and horizon configuration.

B. Comparison with multi-objective baselines

Table IV reports the main comparison. On the two larger instances RA-TGEA dominates: hypervolume 0.971 against 0.890 for the best baseline on I2 (+9.1%) and 0.824 against 0.685 on I3 (+20.4%), both significant at p = 0.014, the smallest attainable p-value with four runs per group. IGD+ improves correspondingly, from 0.084 to 0.041 on I2 and from 0.110 to 0.060 on I3. On the smallest instance I1 the ranking is different: SMS-EMOA attains the best hypervolume (0.990) and RA-TGEA is statistically indistinguishable from NSGA-III. We report this openly; the pattern is consistent with the mechanism, since the benefit of cheap evaluations and gradient-guided moves grows with the dimension of the first stage, which is 37, 65 and 121 variables for I1, I2 and I3. The evaluation-throughput advantage is visible in the last column: within the same 6 s, RA-TGEA performs 1.8 to 3.0 times more evaluations than the baselines.

Figure 3 shows the fronts on I2. RA-TGEA extends noticeably further into the low-cost and low-carbon region, which is where the cost-reducing branch of the local search operates, while remaining competitive at the low-risk end. MOEA/D performs poorly throughout because the penalty treatment of constraint (4) distorts its scalarized aggregation, consistent with the known sensitivity of decomposition-based algorithms to front shape and scalarization [36].

Table IV. Main comparison under an equal wall-clock budget of 6 s (mean +/- sd over 4 seeds)

Inst.	Algorithm	HV	IGD+	Evals	p
I1	RA-TGEA	0.937+/-0.015	0.028	16247	-
	NSGA-III	0.938+/-0.010	0.030	9236	0.557
	NSGA-II	0.971+/-0.008	0.021	9828	1.000
	MOEA/D	0.206+/-0.042	0.525	5028	0.014
	SMS-EMOA	0.990+/-0.018	0.012	9259	1.000
	C-TAEA	0.891+/-0.036	0.071	8326	0.029
I2	RA-TGEA	0.971+/-0.005	0.041	13822	-
	NSGA-III	0.775+/-0.101	0.132	6666	0.014
	NSGA-II	0.835+/-0.032	0.104	6757	0.014
	MOEA/D	0.268+/-0.091	0.501	4072	0.014
	SMS-EMOA	0.722+/-0.067	0.161	6688	0.014
	C-TAEA	0.890+/-0.042	0.084	5688	0.014
I3	RA-TGEA	0.824+/-0.023	0.060	10218	-
	NSGA-III	0.641+/-0.047	0.125	3367	0.014
	NSGA-II	0.685+/-0.017	0.110	3322	0.014
	MOEA/D	0.269+/-0.064	0.468	2480	0.014
	SMS-EMOA	0.629+/-0.033	0.139	3776	0.014
	C-TAEA	0.665+/-0.044	0.136	3572	0.014

p-values are one-sided Mann-Whitney U tests of RA-TGEA against each baseline on hypervolume; 0.014 is the smallest value attainable with 4 runs per group.

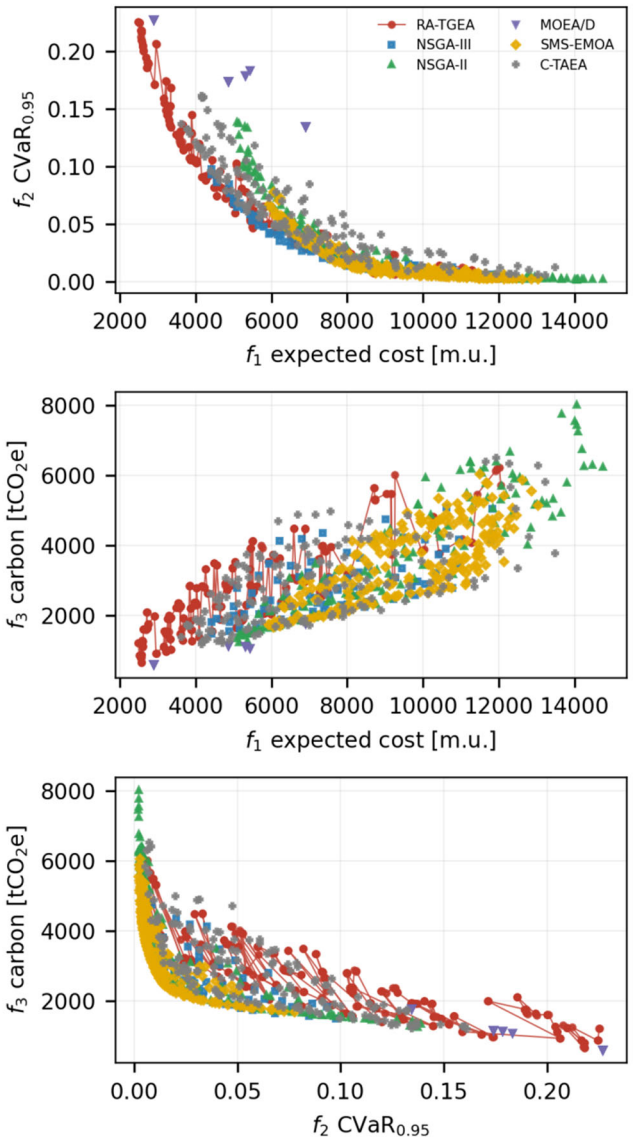

Figure 3. Non-dominated fronts on instance I2 pooled over 4 seeds, projected on the three objective pairs. The proposed method extends further into the low-cost and low-carbon regions.

C. Anytime behaviour and equal evaluation budget

Figure 4(a) shows hypervolume against wall-clock time on I2: RA-TGEA is ahead of every baseline from the first second onwards and its variance across seeds is the smallest, which matters for a planning tool that is expected to return a usable trade-off surface quickly. Figure 4(b) summarizes final quality per instance. Equalizing the number of evaluations at 12,000 instead of the time budget separates the two mechanisms: RA-TGEA reaches hypervolume 0.949 in 5.8 s, NSGA-III 0.808 in 10.7 s, SMS-EMOA 0.784 in 11.0 s and C-TAEA 0.956 in 12.9 s. At equal evaluations C-TAEA is therefore on par with our method, but needs 2.2 times more wall-clock time to get there; the gradient-guided variation accounts for the quality and the progressive scenario refinement for the speed.

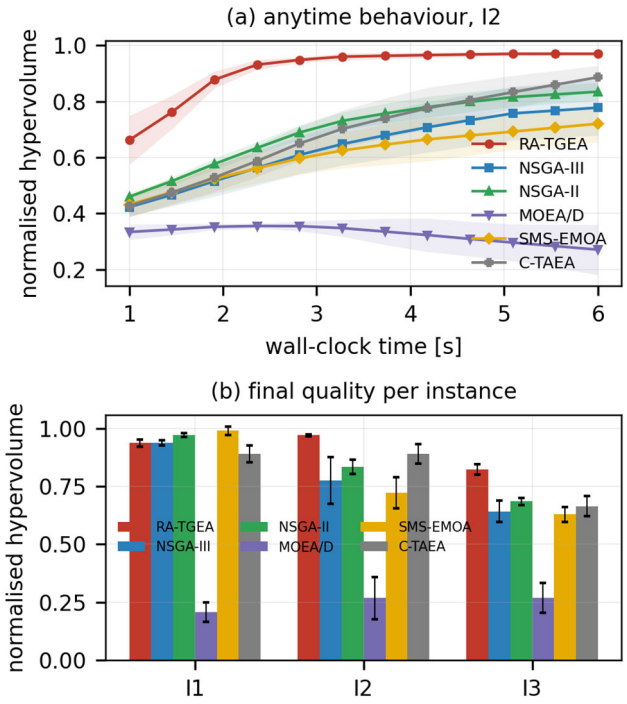

Figure 4. (a) Hypervolume against wall-clock time on I2, mean and standard deviation over 4 seeds. (b) Final hypervolume per instance.

D. Ablation

Table V isolates the three components on I2. Removing the tail-gradient local search is catastrophic, dropping hypervolume from 0.906 to 0.408, because without the cost-reducing withdrawal move the population never reaches the low-cost region of the front at all; the number of evaluations is unchanged, so this is purely a search-quality effect. Removing progressive scenario refinement costs 3.0% of hypervolume and, more tellingly, halves the number of evaluations from 13,764 to 7,082, confirming that refinement buys throughput. Replacing tail-preserving reduction by uniform subsampling costs 1.6% of hypervolume; this effect is in the expected direction but is not statistically significant with four seeds ($p = 0.24$), and we therefore report it as suggestive only.

Table V. Ablation on instance I2 (mean +/- sd over 4 seeds)

Variant	HV	IGD+	Evals	p
RA-TGEA (full)	0.906+/-0.013	0.025	13764	-
without tail-gradient search	0.408+/-0.034	0.384	13680	0.014
without scenario refinement	0.879+/-0.015	0.027	7082	0.057
uniform instead of tail-preserving	0.891+/-0.013	0.028	14154	0.243

E. Value of risk-aware modelling

We next ask what the risk-averse stochastic formulation buys, independently of the solver. Four planning models are optimized with the same solver and then evaluated on 2,000 independent out-of-sample scenarios: the proposed CVaR model, a risk-neutral model that replaces (8) by E[V], a robust model that replaces it by max V, and a deterministic expected-value model solved on a single mean scenario. For each model we record the lowest out-of-sample tail shortfall attainable within a given expected-cost budget.

Table VI reports the result. The deterministic plan is dominated at every budget and, decisively, cannot improve beyond an out-of-sample CVaR of 0.131 no matter how much budget is available: once the mean scenario is covered, the deterministic model sees no reason to buy more, so its entire solution set is blind to the tail. At a budget of 5,000 m.u. the CVaR model attains 0.084 against 0.131, a reduction of 36.1%, and the 99th-percentile shortfall falls from 0.152 to 0.100, a reduction of 34.0%. Among the three stochastic variants the differences are small and not uniform: the CVaR model is best in the tightly budgeted regime (0.120 against 0.126 and 0.128 at a budget of 4,000 m.u., and likewise on the 99th percentile), while at looser budgets the risk-neutral and robust variants are marginally better. The honest conclusion is that the dominant effect is stochastic versus deterministic modelling, and that the choice of tail functional matters mainly when capital is scarce, which is precisely the situation in which accelerator supply is allocated.

Table VI. Out-of-sample tail shortfall attainable within a cost budget (2,000 unseen scenarios, instance I2)

Budget	CVaR	Risk-neutral	Robust	Determ.
4000	0.1198	0.1262	0.1281	0.1383
4500	0.1082	0.1020	0.0978	0.1309
5000	0.0836	0.0811	0.0797	0.1309
5500	0.0776	0.0676	0.0729	0.1309
6000	0.0621	0.0641	0.0563	0.1309
P99 at 5000	0.1000	0.0954	0.0949	0.1515

Entries are the minimum out-of-sample CVaR (0.95) of the weighted service shortfall over all solutions of that model whose out-of-sample expected cost does not exceed the budget.

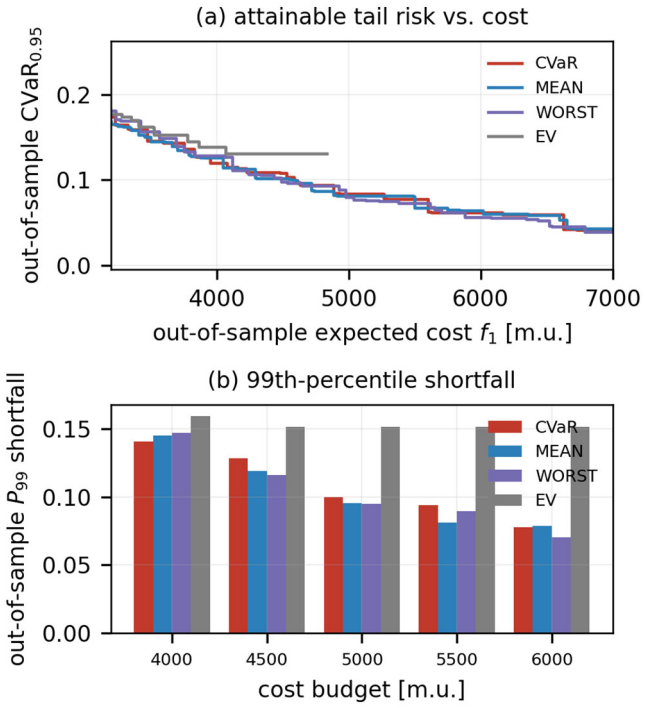

Figure 5. Out-of-sample performance of four planning models on 2,000 unseen scenarios. (a) Attainable tail risk against expected cost. (b) 99th-percentile shortfall by cost budget.

F. Sensitivity and managerial insights

The frontier itself carries the decision-relevant information. Starting from the minimum-cost plan of 2,547 m.u., attaining a tail-shortfall target of 0.20 costs 5.7% more, 0.15 costs 25.4% more, 0.10 costs 59.0% more, 0.05 costs 115.4% more and 0.03 costs 161.2% more. The price of resilience is therefore strongly convex: the last increments of tail protection are bought at several times the marginal cost of the first ones, which argues for setting service targets from this curve rather than from a fixed rule.

Raising the CVaR level from 0.80 to 0.99 moves the knee solution from 3,455 to 4,576 procured units (+32.4%), from four to six qualified suppliers, and lowers the Herfindahl concentration index of the supply portfolio from 0.375 to 0.211: tail aversion is expressed as much through diversification as through volume. Increasing demand-growth uncertainty sd_g from 0.01 to 0.05 raises knee procurement from 2,877 to 4,438 units (+54.3%) and reduces the share allocated to low-carbon suppliers from 1.000 to 0.678, an explicit and quantified erosion of the sustainability objective under greater uncertainty. Doubling disruption intensity adds a supplier and shifts weight towards reliable and zero-lead-time vendors, whose portfolio share rises from 0.000 to 0.079 at the knee. Tightening the share cap from unrestricted to 0.25 raises knee cost from 5,393 to 4,981 m.u. only by reducing volume, and worsens attainable tail risk from 0.072 to 0.087, quantifying the cost of a diversification mandate.

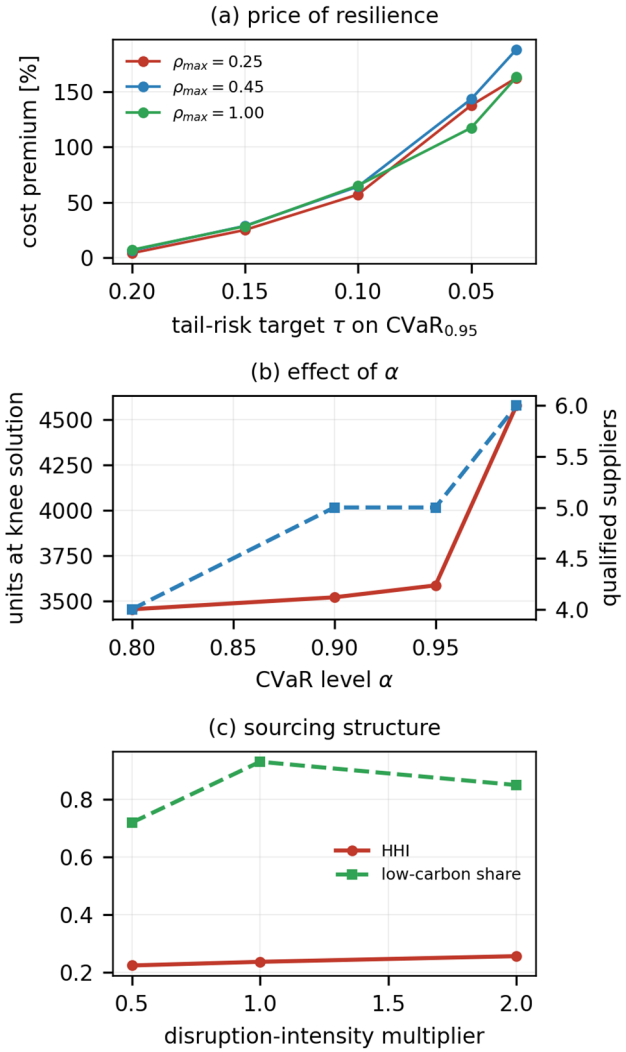

Figure 6. Sensitivity of the risk-efficient frontier. (a) Price of resilience under three supplier-share caps. (b) Effect of the CVaR level on knee-solution volume and supply-base breadth. (c) Sourcing structure against disruption intensity.

7. Discussion and Limitations

Three limitations qualify these results. First, the second stage is evaluated under a parameterized operating policy rather than with fully flexible recourse; by Proposition 1 this is exact for beta = 0 and conservative otherwise, so the reported fronts are inner approximations and the MILP reference front is an outer one. Second, only the demand process is estimated from real data; supplier reliability, disruption and carbon parameters are drawn from literature-consistent ranges rather than from proprietary records, and the mapping of trace statistics estimated at a 72-hour aggregation onto monthly procurement periods is an assumption that the volatility sensitivity partially probes but does not eliminate. Third, the computational budget is deliberately small: 6 s per run and 4 seeds per configuration were chosen to make the full factorial affordable on a single core, which limits the statistical resolution of the tests to p = 0.014 and means the ablation effect of tail-preserving reduction remains suggestive. Larger budgets would sharpen, not reverse, the ranking, since the throughput advantage of the method is budget-independent.

The results also delimit where the method is useful. On small instances an epsilon-constraint MILP is a perfectly good tool and should be preferred when an exact certificate is wanted. The matheuristic earns its place when a dense trade-off surface is needed over many scenarios, which is the operational regime for a planner who wants to see the price of resilience rather than a single plan.

8. Conclusion

We formulated AI-infrastructure supply allocation as a two-stage risk-averse three-objective stochastic program over cost, tail service risk and carbon, with demand uncertainty calibrated from a real production GPU-cluster trace and supplier uncertainty comprising yield, correlated regional disruption and lead-time slippage. The proposed RA-TGEA solver exploits the analytic structure of the risk-averse recourse three times over: an exact closed-form evaluator, an exact closed-form CVaR tail sub-gradient that drives a direction-aware local search, and an unbiased tail-preserving scenario reduction that makes progressive refinement possible. It attains 98.1% of an exact MILP reference hypervolume while producing a dense front about three orders of magnitude faster, and it outperforms five established multi-objective evolutionary algorithms by 9.1% and 20.4% of hypervolume on the two larger instances under an equal time budget. Out of sample, risk-aware stochastic planning reduces attainable tail shortfall by up to 36.1% relative to a deterministic plan of the same cost, and the resulting frontier quantifies a strongly convex price of resilience. Natural extensions are multi-stage recourse with contract renegotiation, distributionally robust ambiguity sets around the bootstrapped demand law, and joint optimization of procurement with carbon-aware workload placement.

References

[1] Sevilla, J., Heim, L., Ho, A., Besiroglu, T., Hobbhahn, M., & Villalobos, P. (2022). Compute trends across three eras of machine learning. In 2022 International Joint Conference on Neural Networks (IJCNN). https://doi.org/10.1109/IJCNN55064.2022.9891914.

[2] Weng, Q., Xiao, W., Yu, Y., Wang, W., Wang, C., He, J., Li, Y., Zhang, L., Lin, W., & Ding, Y. (2022). MLaaS in the wild: Workload analysis and scheduling in large-scale heterogeneous GPU clusters. In Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI) (pp. 945–960).

[3] Jeon, M., Venkataraman, S., Phanishayee, A., Qian, J., Xiao, W., & Yang, F. (2019). Analysis of large-scale multi-tenant GPU clusters for DNN training workloads. In Proceedings of the USENIX Annual Technical Conference (ATC).

[4] Hu, Q., Sun, P., Yan, S., Wen, Y., & Zhang, T. (2021). Characterization and prediction of deep learning workloads in large-scale GPU datacenters. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC).

[5] Snyder, L. V., Atan, Z., Peng, P., Rong, Y., Schmitt, A. J., & Sinsoysal, B. (2016). OR/MS models for supply chain disruptions: A review. IIE Transactions, 48(2), 89–109.

[6] Tomlin, B. (2006). On the value of mitigation and contingency strategies for managing supply chain disruption risks. Management Science, 52(5), 639–657. https://doi.org/10.1287/mnsc.1060.0515

[7] Sawik, T. (2011). Selection of supply portfolio under disruption risks. Omega, 39(2), 194–208. https://doi.org/10.1016/j.omega.2010.06.007

[8] Sawik, T. (2013). Selection of resilient supply portfolio under disruption risks. Omega, 41(2), 259–269. https://doi.org/10.1016/j.omega.2012.05.003

[9] Sawik, T. (2021). On the risk-averse selection of resilient multi-tier supply portfolio. Omega, 101, 102267. https://doi.org/10.1016/j.omega.2021.102267

[10] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 3645–3650). https://doi.org/10.18653/v1/P19-1355

[11] Gupta, U., Kim, Y. G., Lee, S., Tse, J., Lee, H.-H. S., Wei, G.-Y., Brooks, D., & Wu, C.-J. (2021). Chasing carbon: The elusive environmental footprint of computing. In Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA) (pp. 854–867). https://doi.org/10.1109/HPCA51647.2021.00076

[12] Gupta, U., Elgamal, M., Hills, G., Wei, G.-Y., Lee, H.-H. S., Brooks, D., & Wu, C.-J. (2022). ACT: Designing sustainable computer systems with an architectural carbon modeling tool. In Proceedings of the 49th Annual International Symposium on Computer Architecture (ISCA) (pp. 784–799). https://doi.org/10.1145/3470496.3527408

[13] Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon emissions and large neural network training. arXiv:2104.10350.

[14] Masanet, E., Shehabi, A., Lei, N., Smith, S., & Koomey, J. (2020). Recalibrating global data center energy-use estimates. Science, 367(6481), 984–986. https://doi.org/10.1126/science.aba3758

[15] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. Journal of Risk, 2(3), 21–41. https://doi.org/10.21314/jor.2000.038

[16] Artzner, P., Delbaen, F., Eber, J.-M., & Heath, D. (1999). Coherent measures of risk. Mathematical Finance, 9(3), 203–228. https://doi.org/10.1111/1467-9965.00068

[17] Shapiro, A., Dentcheva, D., & Ruszczynski, A. (2009). Lectures on stochastic programming: Modeling and theory. SIAM.

[18] Bertsimas, D., & Sim, M. (2004). The price of robustness. Operations Research, 52(1), 35–53. https://doi.org/10.1287/opre.1030.0061

[19] Mohajerin Esfahani, P., & Kuhn, D. (2018). Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. Mathematical Programming, 171(1–2), 115–166. https://doi.org/10.1007/s10107-017-1172-1

[20] Karabuk, S., & Wu, S. D. (2003). Coordinating strategic capacity planning in the semiconductor industry. Operations Research, 51(6), 839–849. https://doi.org/10.1287/opre.51.6.839.24924

[21] Barahona, F., Bermon, S., Gunluk, O., & Hood, S. (2005). Robust capacity planning in semiconductor manufacturing. Naval Research Logistics, 52(5), 459–468. https://doi.org/10.1002/nav.20087

[22] Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation, 18(4), 577–601. https://doi.org/10.1109/TEVC.2013.2281253

[23] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182–197. https://doi.org/10.1109/4235.996017

[24] Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. Complex Systems, 9, 115–148.

[25] Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal on Optimization, 8(3), 631–657. https://doi.org/10.1137/S1052623494264555

[26] Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 11(6), 712–731. https://doi.org/10.1109/TEVC.2007.892759

[27] Beume, N., Naujoks, B., & Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research, 181(3), 1653–1669. https://doi.org/10.1016/j.ejor.2006.08.008

[28] Li, K., Chen, R., Fu, G., & Yao, X. (2019). Two-archive evolutionary algorithm for constrained multiobjective optimization. IEEE Transactions on Evolutionary Computation, 23(2), 303–315. https://doi.org/10.1109/TEVC.2018.2866870

[29] Blank, J., & Deb, K. (2020). pymoo: Multi-objective optimization in Python. IEEE Access, 8, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2997906

[30] Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation, 3(4), 257–271. https://doi.org/10.1109/4235.797969

[31] Ishibuchi, H., Masuda, H., Tanigaki, Y., & Nojima, Y. (2015). Modified distance calculation in generational distance and inverted generational distance. In Evolutionary Multi-Criterion Optimization (EMO), LNCS 9019 (pp. 110–125). Springer. https://doi.org/10.1007/978-3-319-15892-1_8

[32] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2

[33] Huangfu, Q., & Hall, J. A. J. (2018). Parallelizing the dual revised simplex method. Mathematical Programming Computation, 10(1), 119–142. https://doi.org/10.1007/s12532-017-0088-9

[34] Kleywegt, A. J., Shapiro, A., & Homem-de-Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization, 12(2), 479–502. https://doi.org/10.1137/S1052623400370275

[35] Dupacova, J., Growe-Kuska, N., & Romisch, W. (2003). Scenario reduction in stochastic programming: An approach using probability metrics. Mathematical Programming, 95(3), 493–511. https://doi.org/10.1007/s10107-002-0310-2

[36] Ishibuchi, H., Setoguchi, Y., Masuda, H., & Nojima, Y. (2017). Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. IEEE Transactions on Evolutionary Computation, 21(2), 169–190. https://doi.org/10.1109/TEVC.2016.2598764